This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF THE CLAIMS:

1-2. (Canceled)

3. (Currently Amended) A method for detecting an attack on a data processing system installed on a kernel layer, the method comprising, in the data processing system installed on the kernel layer:

providing an initial secret;

binding the initial secret to data indicative of an initial state of the system, which is installed on the kernel layer between a hardware layer and an operating system layer, via a cryptographic function;

recording state changing administrative actions performed on the system in a log, the state changing administrative actions comprising one or more of: an installation of kernel modules and an alternation of system run-level codes;

prior to performing each state changing administrative action, generating a new secret by performing the cryptographic function on a combination of data indicative of the administrative action and the previous secret, and erasing the previous secret;

evolving the initial secret based on the log to produce an evolved secret;

comparing the evolved secret with the new secret;

determining that the system is uncorrupted if the comparison indicates a match between the evolved secret and the new secret; and

determining that the system ~~in~~ is corrupted if the comparison indicates a mismatch between the evolved secret and the new secret,

wherein the cryptographic function comprises a one-way hash function and the hash function comprises an exponentiation function.

4. (Previously Presented) The method as claimed in claim 3, wherein the cryptographic function comprises a public/private key pair.

5. (Previously Presented) The method as claimed in claim 3, further comprising:

receiving the initial secret from a system administrator.

6-7. (Canceled)

8. (Currently Amended) A data processing system, which is installed on a kernel layer, comprising:

a processor;

a memory connected to the processor; and

detection logic connected to the processor and the memory, the detection logic, in use:

providing an initial secret;

binding the initial secret to data indicative of an initial state of the system, which is installed on the kernel layer between a hardware layer and an operating system layer, via a cryptographic function;

recording state changing administrative actions performed on the system in a log, the state changing administrative actions comprising one or more of: an installation of kernel modules and an alternation of system run-level codes;

prior to performing each state changing administrative action, generating a new secret by performing the cryptographic function on a combination of data indicative of the administrative action and the previous secret, and erasing the previous secret;

evolving the initial secret based on the log to produce an evolved secret;

comparing the evolved secret with the new secret;

determining that the system is uncorrupted if the comparison indicates a match between the evolved secret and the new secret; and

determining that the system ~~in~~ is corrupted if the comparison indicate a mismatch between the evolved secret and the new secret,

wherein the cryptographic function comprises a one-way hash function and the hash function comprises an exponentiation function.

9. (Previously Presented) The system as claimed in claim 8, wherein the cryptographic function comprises a public/private key pair.

10. (Previously Presented) The system as claimed in claim 8, wherein the detection logic receives the initial secret from a system administrator.

11. (Previously Presented) A computer program element comprising computer program code means which, when loaded in a processor of a computer system, configures the processor to perform a method as claimed in claim 3.

12. (Canceled)

13. (Previously Presented) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform method steps for detecting an attack on a data processing system, said method steps comprising the steps of claim 3.

14. (Previously Presented) A computer program product comprising a computer usable medium having computer readable program code means embodied therein for causing a data processing system, the computer readable program code means in said computer program product comprising computer readable program code means for causing a computer to effect the functions of claim 8.

15. (Currently Amended) A method for cryptographic entangling of state and administration in a data processing system installed on a kernel layer, the method comprising:

initializing the system, which is installed on the kernel layer between a hardware layer and an operating system layer, by generating an initial secret releasing binding data;

binding the binding data to the initial secret via a cryptographic function;

updating the initial secret in advance of an administrative action by computing a new secret, the administrative action comprising one or more of: an installation of kernel modules and an alternation of system run-level codes;

erasing the initial secret together with any information from which the initial secret might be derived;

recording data indicative of the administrative action; permitting execution of the administrative action; and

offering a proof that the new secret corresponds to the initial secret as it has evolved according to a record of administrative actions,

wherein the cryptographic function comprises a one-way hash function and the hash function comprises an exponentiation function.

16. (Previously Presented) The method as recited in claim 15, wherein the step of offering retrieves the initial secret via a request for entry of the initial secret by a system administrator, retrieving the record of administrative actions previous stored; and

evolving a candidate secret for the initial secret based on the record of administrative actions retrieved;

comparing the candidate secret with a current secret;

if the candidate secret matches the current secret, reporting that the data processing system is still in an uncorrupted state, and

if the candidate secret does not match the current secret, reporting that the data processing system is in a potentially compromised state.

17. (Previously Presented) The method as recited in claim 15, further comprising permitting detection of any Trojan horse within the system.

18. (Previously Presented) The method as recited in claim 15, wherein the initial secret is supplied via a secure communication channel.

19. (Previously Presented) The method as recited in claim 15, wherein the binding data takes different forms depending on an application of the data processing system, and a trust mechanisms associated with communication of the initial secret.

20. (Canceled)

21-22. (Canceled)

23. (New) The method as recited in claim 15, wherein the step of computing the new secret includes applying a one way function to a combination of a previous secret and data indicative of the administrative action.